

I Claim:

1. A structure for encapsulating a message to be exchanged between an IP phone and an entity within an Ethernet-based PBX, comprising a Protocol Header and an IP
 5 Message body, wherein the Protocol Header includes an indication of Protocol Type for denoting whether the message is an IP message or an encapsulated non-IP message, Device Number for denoting by means of a MAC (Media Access Control) an address for said entity within said PBX to which said message is to be transmitted or from which said message is to be received, and Message Type for identifying the
 10 type of message contained in the IP Message Body.

2. The structure of claim 1, further characterized as follows:

```

15 typedef struct _JPSP_MSG {
    PROTOCOL_HEADER_MSG hdr;
    union _msg {
        MINET_WRAPPER_MSG      MWM;
        DEVICE_REGISTRATION_MSG DRM;
        DEVICE_REGISTRATION_ACK_MSG DRAM;
        DEVICE_UNREGISTER_MSG    DUM;
        DEVICE_UNREGISTER_ACK_MSG DUAM;
        OPEN_RX_STREAM_REQUEST_MSG ORSRM;
        OPEN_RX_STREAM_ACK_MSG    ORSAM;
        CLOSE_RX_STREAM_REQUEST_MSG CRSRM;
        CLOSE_RX_STREAM_ACK_MSG    CRSAM;
        OPEN_TX_STREAM_REQUEST_MSG OTSRM;
        OPEN_TX_STREAM_ACK_MSG     OTSAM;
        CLOSE_TX_STREAM_REQUEST_MSG CTSRM;
        CLOSE_TX_STREAM_ACK_MSG    CTSAM;
        APPLY_TONE_REQUEST_MSG     ATRM;
        REMOVE_TONE_REQUEST_MSG    RTRM;
        DEVICE_PING_REQUEST_MSG     DPRM;
        DEVICE_PING_ACK_MSG         DPAM;
        DEVICE_IP_UPDATE_REQUEST_MSG DIURM;
        DEVICE_IP_UPDATE_ACK_MSG    DIUAM;
    } msg;
} JPSP_MSG;

40 typedef struct {
    protocolType_t  protoType;
    deviceNumber_t  devNum;
    messageType_t  msgType;
} PROTOCOL_HEADER_MSG.

```

45 3. A Device Registration request message sent from the IP Phone in accordance with the structure of claim 1, characterized as follows:

ProtoType = MITEL_INTERNAL
 DevNum = N where N=0,1,2,...,n
 msgType = DEVICE_REGISTRATION

5

devId: 6 unsigned byte array
 mac_addr[6] MAC address of Phone.

10

Note that due to long word alignment, there may be 2 bytes of filler between the MAC address and the next defined field.

devType: 4 bytes , unsigned long integer, Type of device (i.e., SET, PKM, ...)

devNumber: 4 bytes , unsigned long integer, Number of device: Master, Slave01, Slave02, ...

15

ipAddress: structure

ip_addr 4 bytes , unsigned long integer, IP Address of device,

ip_port 2 bytes , unsigned short integer, port number of protocol medium.

20

Note that due to long word alignment, there may be two bytes of filler between this field and the next.

DeviceCaps: structure: Functionality supported by this device

25

strmCodec 4 bytes, unsigned long integer (bitmap), System selected CODEC to use. Multiple CODECs may be logically Ored into this field.

numTxStreams: 4 bytes , unsigned long integer, Number of Tx streams supported by the device

30

numRxStreams: 4 bytes , unsigned long integer, Number of Rx streams supported by the device

prefStrmFrameSizeInMS: 4 bytes , unsigned long integer, Devices preferred frame size for streams (in ms)

35

silenceSupp: 4 bytes , unsigned long integer:
 silenceSupp=0: device does not support silence suppression
 silenceSupp=1: device supports silence suppression

40

toneGeneration: 4 bytes , unsigned long integer:
 toneGeneration =0: device does not support local tone generation.
 toneGeneration =1: device supports local tone generation.

45

4. A Device Registration request Acknowledgment message sent from the PBX in accordance with the structure of claim 1, characterized as follows:

22

ProtoType = MITEL_INTERNAL

DevNum = N where N=0,1,2,...,n

5 msgType = DEVICE_REGISTRATION_ACK

reqStatus: 4 bytes , unsigned long integer, Success/Failure Result of the request

sysToken: 4 bytes , unsigned long integer, System defined "token" that must be passed
back with any follow up message related to this message i.e. Device
Unregister.

10

5. A Device De-Registration Request message sent from the IP Phone in
accordance with the structure of claim 1, characterized as follows:

15 ProtoType = MITEL_INTERNAL

DevNum = N where N=0,1,2,...,n

msgType = DEVICE_DEREGISTRATION

20 sysToken: 4 bytes , unsigned long integer, System defined "token" taken from the
Registration Acknowledgment from the system.

devType: 4 bytes , unsigned long integer, Type of device (i.e., SET, PKM, etc...)

devNumber: 4 bytes , unsigned long integer, Number of device: Master, Slave01,
Slave02, ...

ipAddress: structure

25

ip_addr 4 bytes , unsigned long integer, IP Address of device,

ip_port 2 bytes , unsigned short integer, port number of protocol medium.

6. A Device De-Registration Acknowledgment message sent from the PBX in
accordance with the structure of claim 1, characterized as follows:

30

ProtoType = MITEL_INTERNAL

DevNum = N where N=0,1,2,...,n

msgType = DEVICE_DEREGISTRATION_ACK

35

reqStatus: 4 bytes , unsigned long integer, Success/Failure Result of the request

devNumber: 4 bytes , unsigned long integer, Number of device: Master, Slave01,
Slave02, ...

wherein

40

devType:

23

5 INVALID_DEVICE_TYPE 0x00000000
 IP_SUPERSET4001 0x00000001
 IP_SUPERSET4015 0x0000009f
 IP_SUPERSET4025 0x000000a0
 IP_SUPERSET4150 0x00000004
 PKM 0x00000005
 AIM 0x00000006
 10 SYMBOL_PROXY 0x00000007
 SYMBOL_SET 0x00000008
 TELEWORKER_PROXY 0x00000009
 TELEWORKER_SET 0x0000000a
 E2T_PROXY 0x0000000b
 15 MAX_DEVICE_TYPE 0x0000000c

devNumbers:

MASTER_DEVICE 0x00000000
 Where Set=0, and any attached devices will be numbered MASTER_DEVICE + n
 20 where n >= 1

reqStatus (Success/failure codes):

25 MTL_SUCCESS 0x00000000
 MTL_FAILURE 0x00000001
 MTL_NO_PERMISSIONS 0x00000002
 MTL_NO_RESOURCES 0x00000003
 MTL_INVALID_DEVICE 0x00000004
 30 MTL_INVALID_REQUEST 0x00000005

devCodecs bitmap:

NO_CODEC_SUPPORT 0x0 (000 00000000)
 G711_ULAW64 0x1 (000 00000001)
 35 G711_ALAW64 0x2 (000 00000010)
 G728 0x4 (000 00000100)
 G729 0x8 (000 00001000)
 G729_ANNEXB 0x10 (000 00010000)
 G729_ANNEXA_w_ANNEXB 0x20 (000 00100000)
 40 G723 0x40 (000 01000000)
 G7231_ANNEXC 0x80 (000 10000000)
 Placeholder1 0x100 (001 00000000)
 Placeholder2 0x200 (010 00000000)
 Placeholder3 0x400 (100 00000000)
 45 INVALID_CODEC 0x7FF (111 11111111).

7. A Device ICMP Echo (Ping) request message to the phone in accordance with the structure of claim 1, characterized as follows:

5 ProtoType = MITEL_INTERNAL
 DevNum = N where N=0,1,2,...,n
 msgType = DEVICE_PING

	hostIpAddress:	structure
10	ip_addr	4 bytes , unsigned long integer, IP Address of device to PING,
	ip_port	2 bytes , unsigned short integer, port number is IGNORED.
15		Note that due to long word alignment, there may be two bytes of filler following this field.
	numRequests	4 bytes , unsigned long integer, Number of ping requests to send
20	pktSize	4 bytes , unsigned long integer, Size of data packet to send (in bytes)
	pktDelay	4 bytes , unsigned long integer, Inter packet delay in Milliseconds
	timeOut	4 bytes , unsigned long integer, Ping request timeout in Milliseconds
25	qosLevel	4 bytes , unsigned long integer, QOS level requested.

8. A Device ICMP Echo (Ping) results message sent from the phone to the PBX in accordance with the structure of claim 1, characterized as follows:

30 ProtoType = MITEL_INTERNAL
 DevNum = N where N=0,1,2,...,n
 msgType = DEVICE_PING_ACK

35	hostIpAddress:	structure
	ip_addr	4 bytes , unsigned long integer, IP Address of device that was PINGed,
40	ip_port	2 bytes , unsigned short integer, port number is IGNORED.
		Note that due to long word alignment, there may be two bytes of filler following this field.

	pktsSent	4 bytes , unsigned long integer, Number of ICMP echo requests sent
	pktsRecv	4 bytes , unsigned long integer, Number of ICMP echo replys received
	pktLoss	4 bytes , unsigned long integer, Percentage of packets lost
5	rttMax	4 bytes , unsigned long integer, Maximum round trip time (in milliseconds)
	rttMin	4 bytes , unsigned long integer, Minimum round trip time (in milliseconds)
	rttAvg	4 bytes , unsigned long integer, Average round trip time (in milliseconds)

and wherein
10 qosLevel:

	QOS_LEVEL_NONE	0xffffffff
	QOS_LEVEL_0	0x00000000
	QOS_LEVEL_1	0x00000001
15	QOS_LEVEL_2	0x00000002
	QOS_LEVEL_3	0x00000003
	QOS_LEVEL_4	0x00000004
	QOS_LEVEL_5	0x00000005
	QOS_LEVEL_6	0x00000006
20	QOS_LEVEL_7	0x00000007

9. An Apply Tone device tone generation request message to the phone in accordance with the structure of claim 1, characterized as follows:

25 ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,...n
msgType = APPLY_TONE

30	sysToken:	4 bytes , unsigned long integer, System defined "token" that must be passed back with the Remove Tone request.
	sysStrmID:	4 bytes , unsigned long integer, System provided stream ID which maps the voice streams to legacy B channels
	tone[MAX_COMPLEX_TONE]:	array of tohe structures of frequencies the DSP is to play
35	on_T1	2 bytes, unsigned long integer, Duration in ms of 1st ON period
	off_T1	2 bytes, unsigned long integer, Duration in ms of 1st OFF period
	on_T2	2 bytes, unsigned long integer, Duration in ms of 2nd ON period
	off_T2	2 bytes, unsigned long integer, Duration in ms of 2nd OFF period
40	num_cycles	2 bytes, unsigned long integer, Number of times to repeat the ON/OFF sequence
	tail	2 bytes, unsigned long integer, After num_cycles, 0 = leave tone off, 1 = on
	freq_1	2 bytes, unsigned long integer, 1st frequency component in Hz
	freq_2	2 bytes, unsigned long integer, 2nd frequency component in Hz
45	level_1	2 bytes, unsigned long integer, 1st frequency signal level

26

	level_2	2 bytes, unsigned long integer, 2nd frequency signal level
	action	2 bytes, unsigned long integer, indicates the action to take on completion of the tone. The actions are either to continue to the next tone descriptor, reconnect to the audio stream, or just stop.
5		Note that due to long word alignment, there may be 2 bytes of filler following this field.
	toneId:	4 bytes, unsigned long integer, System Tone ID of the tone being applied
10	inject;	4 bytes, unsigned long integer, specify whether to inject the tone on top of voice or not. This is unused by the phone since the tone will always take precedence over voice.

10. A Remove Tone device tone generation request message to the phone in accordance with the structure of claim 1, characterized as follows:

15	ProtoType = MITEL_INTERNAL	
	DevNum = N where N=0,1,2,...,n	
	msgType = REMOVE_TONE	
20	sysToken:	4 bytes, unsigned long integer, System defined "token" that was given with the Apply Tone request.
	sysStrmID:	4 bytes, unsigned long integer, System provided stream ID which maps the voice streams to legacy B channels
25	tone[MAX_COMPLEX_TONE]:	array of tone structures of frequencies the DSP was playing out to the CODEC that it is to remove. Note that this is IGNORED BY IP PHONE
	on_T1	2 bytes, unsigned long integer, Duration in ms of 1st ON period
30	off_T1	2 bytes, unsigned long integer, Duration in ms of 1st OFF period
	on_T2	2 bytes, unsigned long integer, Duration in ms of 2nd ON period
	off_T2	2 bytes, unsigned long integer, Duration in ms of 2nd OFF period
	num_cycles	2 bytes, unsigned long integer, Number of times to repeat the ON/OFF sequence
35	tail	2 bytes, unsigned long integer, After num_cycles, 0 = leave tone off, 1 = on
	freq_1	2 bytes, unsigned long integer, 1st frequency component in Hz
	freq_2	2 bytes, unsigned long integer, 2nd frequency component in Hz
	level_1	2 bytes, unsigned long integer, 1st frequency signal level
40	level_2	2 bytes, unsigned long integer, 2nd frequency signal level
	action	2 bytes, unsigned long integer, indicates the action to take on completion of the tone. The actions are either to continue to the next tone descriptor, reconnect to the audio stream, or just stop.
45	and wherein	
	inject:	

27

	NOT_INJECTED	0x00000000
	NORMAL_INJECTION	0x00000001
	MAX_TONE_INJECT	0x00000002
5	MAX_COMPLEX_TONE	3

action:		
	NEXT	0x00000000
	RECONNECT	0x00000001
10	STOP	0x00000002.

11. An Open Receive Stream Request to the phone in accordance with the structure of claim 1, characterized as follows:

15 ProtoType = MITEL_INTERNAL
 DevNum = N where N=0,1,2,...n
 msgType = OPEN_RX_STREAM

20	sysToken:	4 bytes, unsigned long integer, System defined "token" that must be passed back with the corresponding Close Receive Stream Request .
	sysStrmID:	4 bytes, unsigned long integer, System provided stream ID. This field denotes the B channel the connection should assume.
25	strmCodec	4 bytes, unsigned long integer (bitmap), System selected CODEC to use. Multiple CODECs may be logically Ored into this field.
	strmFrameSizeInMS	4 bytes, unsigned long integer, Preferred CODEC frame size for the RX stream (in milliseconds)
30	isMulticast	4 bytes, unsigned long integer isMulticast =0: no Multicast, ignore mcIpAddress. isMulticast =1: the stream must be bound to the mcIpAddress Multicast address.

35	mcIpAddress:	structure
	ip_addr	4 bytes, unsigned long integer, Multicast address to receive on
	ip_port	2 bytes , unsigned short integer, Multicast port number to receive on.

40 Note that due to long word alignment, there may be two bytes of filler following this field.

45	SrcIpAddress:	structure: IGNORED BY THE IP PHONE.
	ip_addr	4 bytes, unsigned long integer, The ip address of the device that will be transmitting to the phone.

ip_port

2 bytes , unsigned short integer, port number used by the device that will be transmitting to the phone.

Note that due to long word alignment, there may be two bytes of filler following this field.

noSilence

4 bytes, unsigned long integer,

noSilence = 0: no silence suppression applied by the transmitting end

noSilence=1: silence suppression is being applied by the transmitting end

12. An Open Receive Stream Acknowledgement from the IP Phone to the PBX in accordance with the structure of claim 1, characterized as follows:

ProtoType = MITEL_INTERNAL

$$\text{DevNum} = N \text{ where } N=0,1,2,\dots,n$$

```
msgType = OPEN_RX_STREAM_ACK
```

reqStatus:

4 bytes, unsigned long integer, Success/Failure Result of the request

sysToken:

4 bytes, unsigned long integer, System provided "token"
from the request message

rxConnectionID:

4 bytes, unsigned long integer, Device selected stream/connection identifier. The IP Phone returns the value of the sysStreamID (B channel) in this field structure

rxStrmIpAddress:

ip_addr

4 bytes, unsigned long integer, The local ip address that will receive stream

ip_port

2 bytes , unsigned short integer, local port number to receive on.

13. A Close Receive Stream Request from the PBX to the IP Phone in accordance with the structure of claim 1, characterized as follows:

ProtoType = MITEL_INTERNAL

$$\text{DevNum} = N \text{ where } \bar{N} = 0, 1, 2, \dots, n$$

```
msgType = CLOSE_RX_STREAM
```

sysToken:

4 bytes, unsigned long integer, System defined "token" that was given with the Open Receive Stream Request .

sysStrmID:

4 bytes, unsigned long integer, Id of RX stream/connection (B channel) to close.

14. A Close Receive Stream Acknowledgement from the IP Phone in accordance with the structure of claim 1, characterized as follows:

5 ProtoType = MITEL_INTERNAL
 DevNum = N where N=0,1,2,...,n
 msgType = CLOSE_RX_STREAM_ACK

10	reqStatus:	4 bytes, unsigned long integer, Success/Failure Result of the request
	sysToken:	4 bytes, unsigned long integer, System provided "token" from the request message
	rxStrmStats:	structure: Stream statistics upon closure
15	Packets.recv	4 bytes, unsigned long integer, number of RTP packets received
	Bytes.recv	4 bytes, unsigned long integer, number of voice octets received
	Errors.rxStream	4 bytes, unsigned long integer, number of RTP errors received
20	Jitter.rxStream	4 bytes, unsigned long integer, estimate of average jitter over duration of call.
	Duration.rxStream	4 bytes, unsigned long integer, duration of call in seconds
	IpAddress.src:	structure
25	ip_addr	4 bytes, unsigned long integer, the local ip address
	ip_port	2 bytes, unsigned short integer, the local port number.

15. An Open Transmit Stream Request to the IP Phone in accordance with the structure of claim 1, characterized as follows:

30 ProtoType = MITEL_INTERNAL
 DevNum = N where N=0,1,2,...,n
 msgType = OPEN_TX_STREAM

35	sysToken:	4 bytes, unsigned long integer, System defined "token" that must be passed back with the corresponding Close Transmit Stream Request.
	sysStrmID:	4 bytes, unsigned long integer, System provided stream ID. This field denotes the B channel the connection should assume.
40	strmCodec	4 bytes, unsigned long integer (bitmap), System selected CODEC to use. Multiple CODECs may be logically Ored into this field.
	strmFrameSizeInMS	4 bytes, unsigned long integer, Preferred CODEC frame size for the TX stream (in milliseconds)
45	destStrmIpAddress:	structure

	ip_addr	4 bytes, unsigned long integer, The IP address of the device to transmit to.
	ip_port	2 bytes, unsigned short integer, port number used by the device that will be transmitting to the phone.
5		Note that due to long word alignment, there may be two bytes of filler following this field.
10	qosLevel	4 bytes, unsigned long integer, QoS level requested. If 0xffffffff, then no 802.1Q tag, else if 0-7, assume 802.1Q tag and set priority field to the qosLevel
	noSilence	4 bytes, unsigned long integer noSilence = 0: disable silence suppression on the Tx stream noSilence = 1: enable silence suppression on the Tx stream.
15		

16. An Open Transmit Stream Acknowledgement from the IP Phone in accordance with the structure of claim 1, characterized as follows:

20

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,...,n
msgType = OPEN_TX_STREAM_ACK

25	reqStatus:	4 bytes, unsigned long integer, Success/Failure Result of the request
	sysToken:	4 bytes, unsigned long integer, System provided "token" from the request message
30	txConnectionID:	4 bytes, unsigned long integer, Device selected stream/connection identifier. The IP Phone returns the value of the sysStreamID (B channel) in this field
	txStrmIpAddress:	structure
	ip_addr	4 bytes, unsigned long integer, The local IP address that will transmit stream
35	ip_port	2 bytes, unsigned short integer, local port number the phone will transmit from.

17. A Close Transmit Stream Request to the IP Phone in accordance with the structure of claim 1, characterized as follows:

40

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,...,n
msgType = CLOSE_TX_STREAM

31

sysToken: 4 bytes, unsigned long integer, System defined "token" that was given with the Open Transmit Stream Request .

sysStrmID: 4 bytes, unsigned long integer, Id of TX stream/connection (B channel) to close.

5

18. A Close Transmit Stream Acknowledgement from the IP Phone in accordance with the structure of claim 1, characterized as follows:

ProtoType = MITEL_INTERNAL
 10 DevNum = N where N=0,1,2,...,n
 msgType = CLOSE_TX_STREAM_ACK

	reqStatus:	4 bytes, unsigned long integer, Success/Failure Result of the request
15	sysToken:	4 bytes, unsigned long integer, System provided "token" from the request message
	txStrmStats:	structure: Stream statistics upon closure
	Packets.sent	4 bytes, unsigned long integer, number of RTP packets sent
20	Bytes.sent	4 bytes, unsigned long integer, number of voice octets sent
	Errors.txStream	4 bytes, unsigned long integer, number of RTP errors sent. IGNORE, NOT RELEVANT
25	Jitter.txStream	4 bytes, unsigned long integer, estimate of average jitter over duration of call. IGNORE, NOT RELEVANT
	Duration.txStream	4 bytes, unsigned long integer, duration of call in seconds
	IpAddress.dest:	structure
	ip_addr	4 bytes, unsigned long integer, the local IP address used to Tx
30	ip_port	2 bytes , unsigned short integer, the local port number used to Tx.

and wherein

reqStatus (Success/failure codes):

35	MTL_SUCCESS	0x00000000
	MTL_FAILURE	0x00000001
	MTL_NO_PERMISSIONS	0x00000002
	MTL_NO_RESOURCES	0x00000003
40	MTL_INVALID_DEVICE	0x00000004
	MTL_INVALID_REQUEST	0x00000005

45 SysStrmID:

IP Set Stream IDs: (NOTE: TX is always even) used for sysStrmID of Tx & Rx connect requests

	STREAM_ID_IP_SET_TX_1	0x00000000	// B1 TX
5	STREAM_ID_IP_SET_RX_1	0x00000001	// B1 RX
	STREAM_ID_IP_SET_TX_2	0x00000002	// B2 TX
	STREAM_ID_IP_SET_RX_2	0x00000003	// B2 RX

devCodecs bitmap:

10	NO_CODEC_SUPPORT	0x0	(000 00000000)
	G711_ULAW64	0x1	(000 00000001)
	G711_ALAW64	0x2	(000 00000010)
	G728	0x4	(000 00000100)
15	G729	0x8	(000 00001000)
	G729_ANNEXB	0x10	(000 00010000)
	G729_ANNEXA_w_ANNEXB	0x20	(000 00100000)
	G723	0x40	(000 01000000)
	G7231_ANNEXC	0x80	(000 10000000)
20	Placeholder1	0x100	(001 00000000)
	Placeholder2	0x200	(010 00000000)
	Placeholder3	0x400	(100 00000000)
	INVALID_CODEC	0x7FF	(111 11111111)

25 qosLevel:

	QOS_LEVEL_NONE	0xffffffff
	QOS_LEVEL_0	0x00000000
	QOS_LEVEL_1	0x00000001
30	QOS_LEVEL_2	0x00000002
	QOS_LEVEL_3	0x00000003
	QOS_LEVEL_4	0x00000004
	QOS_LEVEL_5	0x00000005
	QOS_LEVEL_6	0x00000006
35	QOS_LEVEL_7	0x00000007.

19. A Device IP address update request message to the phone in accordance with the structure of claim 1, characterized as follows:

40 ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,...,n
msgType = DEVICE_IP_UPDATE

	devNumber	4 bytes , unsigned long integer, Number of device: Master, Slave01, Slave02, ...
	oldIpAddress:	structure
5	ip_addr	4 bytes , unsigned long integer, old IP Address of device
	ip_port	2 bytes , unsigned short integer, old port number of device
		Note that due to long word alignment, there may be two bytes of filler following this field.
10	newIpAddress:	structure
	ip_addr	4 bytes , unsigned long integer, new IP Address of device
	ip_port	2 bytes , unsigned short integer, new port number of device.

15 20. A Device IP address update acknowledgement from the phone in accordance
with the structure of claim 1, characterized as follows:

ProtoType = MITEL_INTERNAL
DevNum = N where N=0,1,2,...,n
20 msgType = DEVICE_IP_UPDATE_ACK

reqStatus: 4 bytes , unsigned long integer, Success/Failure Result of
the request

and wherein

25 **reqStatus (Success/failure codes):**

	MTL_SUCCESS	0x00000000
	MTL_FAILURE	0x00000001
30	MTL_NO_PERMISSIONS	0x00000002
	MTL_NO_RESOURCES	0x00000003
	MTL_INVALID_DEVICE	0x00000004
	MTL_INVALID_REQUEST	0x00000005

35 **devNumbers:**

MASTER_DEVICE 0x00000000
Where Set=0, and any attached devices will be numbered MASTER_DEVICE + n
where n >= 1.

40

21. A Wrapper structure for messages to and from the IP Phone in accordance
with the structure of claim 1, characterized as follows:

34

ProtoType = MINET_MTS22
DevNum = N where N=0,1,2,...,n
msgType = MINET_WRAPPER

5 msgLen: 4 bytes , unsigned long integer, length of the
 following MINET message.
 msg[MAX_MINET_SIZE] array unsigned char, the MTS22 MINET
10 message,
 and wherein

15 MAX_MINET_SIZE 160.

T E R M I N A T E D